МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Архітектура комп'ютера»**

## на тему:«Команди пересилання та обміну даними в мові Асемблера»

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

**Тема.** Команди пересилання та обміну даними в мові Ассемблера.

**Мета.** Вивчити принципи організації пам’яті комп’ютера, адресації даних. Ознайомитись з поняттям ефективної, логічної і фізичної адреси. Вивчити регістри центрального процесора, їх призначення і взаємозв’язок. Вивчити команди пересилання і обміну даними, набути практичних навичок їх використання.

**Постановка задачі**

Розробити програму згідно індивідуального завдання для обробки інформації, що міститься в сегменті даних (колонка «Змінні» в таблиці):

1) змінити текст згідно завданню обробки тексту (колонка «Обробка тексту» в таблиці);

2) записати в пам’ять логічну адресу (сегментну адресу і зміщення) другого числа (колонка «Змінні»);

3) переставити місцями старшу і молодшу частини чисел (колонка «Змінні») у форматі слова (двома способами) і подвійного слова (одним способом). Всі три способи обміну частин чисел повинні бути різними (команди пересилання, команда обміну, стек). Результати записати в нові змінні.



Turbo Assembler Version 3.0 10/23/20 01:55:47 Page 1

lr20.asm

1 ; Розробити програму згідно індивідуального +

2 завдання для обробки інформації,

3 ; що міститься в сегменті даних (колонка «Змінні» в +

4 таблиці)

5

6 ;сегмент стека

7 0000 stk segment stack

8 0000 80\*(05) db 128 dup(5)

9 0080 stk ends

10

11 ;сегмент даних cd

12 0000 data segment para public 'data'

13 0000 45 61 73 79 20 63 6F+ text db "Easy come, easy go."

14 6D 65 2C 20 65 61 73+

15 79 20 67 6F 2E

16 0013 88B9 numbers dw -30535

17 0015 FFF31304 dd -847100

18 0019 8B db -117

19 001A ???????? address dd ?

20 001E ???? a dw ?

21 0020 ???? b dw ?

22 0022 ???????? c dd ?

23 0026 data ends

24

25 ;сегмент команд

26 0000 code segment para public 'code'

27 assume cs:code, ds:data, ss:stk

28 0000 begin:

29 0000 B8 0000s mov ax, data

30 0003 8E D8 mov ds, ax

31 0005 8A 26 0000r mov ah, text+0 ;Прочитаємо E

32 0009 A0 000Br mov al, text+11 ;Прочитаємо e

33 000C 86 E0 xchg ah, al ;Міняємо місцями інформацію в +

34 операндах

35 000E 88 26 0000r mov text+0, ah ;Записуємо е

36 0012 A2 000Br mov text+11, al ;Записуємо Е

37 ; адреса

38 0015 8C 1E 001Ar mov word ptr address, ds

39 0019 B8 0015r lea ax, numbers+2

40 001C A3 001Cr mov word ptr address+2, ax

41 ;1-й способ

42 001F A1 0013r mov ax, numbers

43 0022 88 26 001Er mov byte ptr a, ah

44 0026 A2 001Fr mov byte ptr a+1, al

45 ;2-й способ

46 0029 A1 0013r mov ax, numbers

47 002C 86 C4 xchg al, ah

48 002E A3 0013r mov numbers, ax

49 ;3-й способ

50 0031 FF 36 0015r push word ptr numbers+2

51 0035 FF 36 0017r push word ptr numbers+4

52 0039 8F 06 0022r pop word ptr c

53 003D 8F 06 0024r pop word ptr c+2

54 ;завершення програми

55 0041 B8 4C00 mov ax, 4C00h

56 0044 CD 21 int 21h

57 0046 code ends

\_Turbo Assembler Version 3.0 10/23/20 01:55:47 Page 2

lr20.asm

58 end begin

\_Turbo Assembler Version 3.0 10/23/20 01:55:47 Page 3

Symbol Table

Symbol Name Type Value

??DATE Text "10/23/20"

??FILENAME Text "lr20 "

??TIME Text "01:55:47"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text LR20

@WORDSIZE Text 2

A Word DATA:001E

ADDRESS Dword DATA:001A

B Word DATA:0020

BEGIN Near CODE:0000

C Dword DATA:0022

NUMBERS Word DATA:0013

TEXT Byte DATA:0000

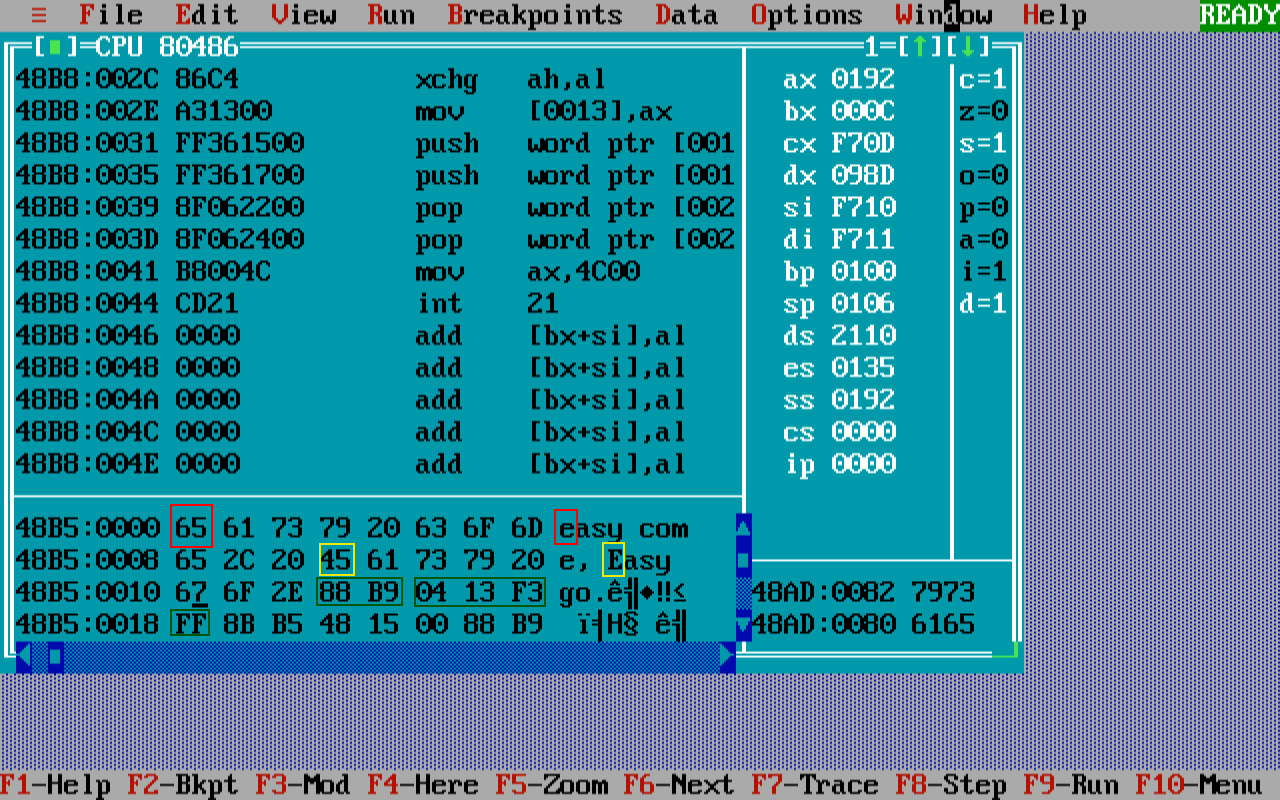
Groups & Segments Bit Size Align Combine Class

CODE 16 0046 Para Public CODE

DATA 16 0026 Para Public DATA

STK 16 0080 Para Stack

\_



**В результаті виконання програми помінялись місцями перші літери першого та третього слова, всі інші літери міняти немає потреби, адже код символів у них абсолютно однаковий. Помінялись місцями старші та молодші частини чисел у форматі слова та подвійного слова. На скріншоті їх машинне представлення обведене зеленими прямокутниками.**

**B9 88 🡺 88B9**

**FF F3 13 04 🡺 04 13 F3 FF**

**Висновок**

Під час виконання лабараторної роботи зрозумів, що можу користуватися регістрами загального призначення на свій розсуд. Перед виконанням кожної програми в асемлері сегмент даних заносить до регістру DS. За допомогою команди lea можна в регістр записувати ефективну адресу операнда. Команда mov копіює дані з одного операнда в інший, проте в неї також обмеження у використанні. Командами pop та push можна класти дані в стек та витягувати їх звідти, працює лише з даними у форматі слова. Команда xcng міняє місцями дані в операндах.